US Army Corps of Engineers®
Engineer Research and Development Center

ERDC
INNOVATIVE SOLUTIONS
for a safer, better world

Military Engineering Applied Research

High-Fidelity Simulations of Electromagnetic Propagation and RF Communication Systems

T53 Final Report

Samuel S. Streeter, Daniel J. Breton, Michele L. Maxson, and Christopher T. Goodin

May 2017
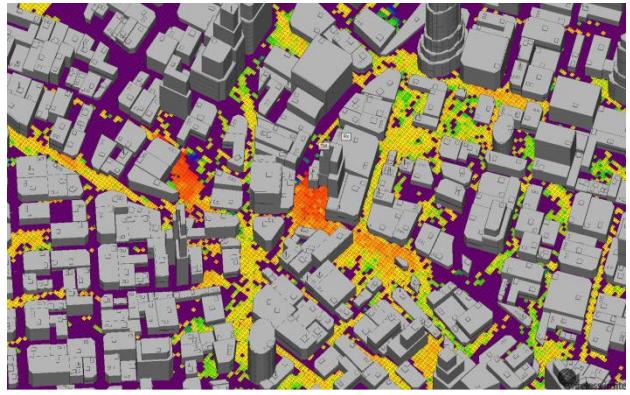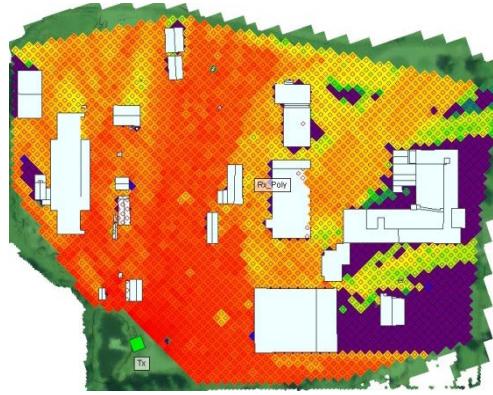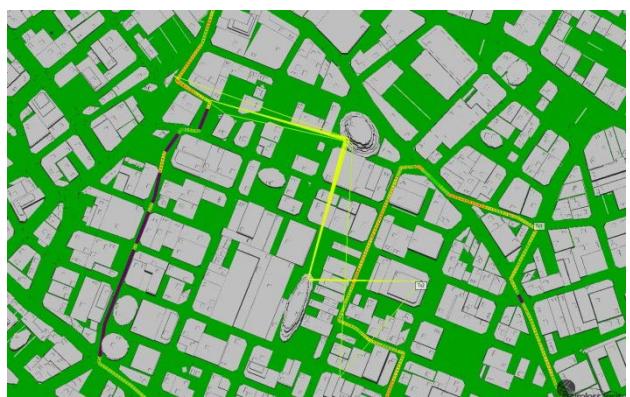
The U.S. Army Engineer Research and Development Center (ERDC) solves the nation's toughest engineering and environmental challenges. ERDC develops innovative solutions in civil and military engineering, geospatial sciences, water resources, and environmental sciences for the Army, the Department of Defense, civilian agencies, and our nation's public good. Find out more at www.erdc.usace.army.mil.

To search for other technical reports published by ERDC, visit the ERDC online library at http://acwc.sdp.sirsi.net/client/default.

High-Fidelity Simulations of Electromagnetic Propagation and RF Communication Systems

T53 Final Report

Samuel S. Streeter, Daniel J. Breton, and Michele L. Maxson

U.S. Army Engineer Research and Development Center (ERDC)
Cold Regions Research and Engineering Laboratory (CRREL)
72 Lyme Road
Hanover, NH 03755-1290

Christopher T. Goodin

U.S. Army Engineer Research and Development Center (ERDC)
Geotechnical and Structures Laboratory (GSL)
3909 Halls Ferry Road
Vicksburg, MS 39180-6199

Final Report

Abstract

Newly developed radio-frequency propagation models estimate signal strength, signal coverage, and bit error rates to support mission planning for robotic platforms operating in urban areas. This study involved high-fidelity modeling on a graphics processing unit workstation and included full three-dimensional analysis of reflection, transmission, and diffraction propagation effects within urban landscapes. Real-time propagation modeling is made possible using an application programming interface (API) with simpler, faster models whose output can, in principle, be used for mission planning or platform performance assessment within a virtual scene. This report presents the results of two test cases—within a virtual rendering of the U.S. Army Cold Region Research and Engineering Laboratory campus and within a fabricated dense urban scene—to demonstrate the ability to generate high-fidelity radio-frequency propagation models from building and terrain data derived from (1) LiDAR (Light Detection and Ranging) and digital elevation models and (2) Virtual Autonomous Navigation Environment (VANE) scenes. This report outlines steps necessary to produce lower fidelity, higher speed models using the API and discusses how the API could interface with existing virtual environments and mission-planning tools.

Contents

Figures and Tables

Figures

Preface

Acronyms and Abbreviations

API	Application Programming Interface
BER	Bit Error Rate
CRREL	Cold Regions Research and Engineering Laboratory
DEM	Digital Elevation Model
EM	Electromagnetic
ERDC	U.S. Army Engineer Research and Development Center
FDTD	Finite-Difference Time-Domain
GIS	Geographic Information System
GPU	Graphics Processing Unit
GSL	Geotechnical and Structure Laboratory
ISM	Industrial, Scientific, and Medical
LiDAR	Light Detection and Ranging
MIMO	Multiple Output
NLOS	Non-Line-of-Sight
RDE	Research and Development Environment
RF	Radio Frequency
Rx	Receiver
3-D	Three-Dimensional
2-D	Two-Dimensional
TIREM	Terrain Integrated Rough Earth Model
Tx	Transmitter
VANE	Virtual Autonomous Navigation Environment

VPUP   Vertical Plane Urban Propagation

VTRPE   Variable Terrain Radio Parabolic Equation

WI   Wireless InSight

1   Introduction

1.1   Background

The radio-frequency (RF) telemetry link between a robotic platform and its human operator is a crucial part of any robotics operation. Such a link carries commands, sensor data, live video, system status information, and other information critical to mission success. The impact of the propagation environment (also known as the RF channel) on signal power and fidelity becomes important, especially for high-data-rate applications, such as robot telemetry. Planning robotics operations with RF telemetry links requires knowledge (or prediction) of the electromagnetic environment not just in the vicinity of the platform itself but over the kilometer-scale distances and complex urban environments separating the operator from the platform. Doing so for ground-based platforms in an urban environment is especially challenging, given the heavily multipathed and shadowed environment and the frequent need to exploit non-line-of-sight (NLOS) channels for successful operations (Parsons 2000; Longley 1978; Suzuki 1977; Kalliola, et al. 2003; Gans 1972; Cox and Leck 1975; Li et al. 2012). RF propagation modeling is required to establish and maintain a robotics telemetry link in an urban environment.

Urban RF propagation models can be broken into two broad groups: empirical and deterministic (Greenberg and Klodzh 2015). Empirical models are typically based on measurements made in a particular environment, within a particular frequency band, and are not geospatially aware. As such, these models have little computational burden, only require basic transmitter (Tx) and receiver (Rx) geometry information, and completely ignore important reflection and shadowing effects generated in urban terrain. Such models are useful for inexpensively estimating path losses within a generic city or within a city whose geometry is unknown. Deterministic models, on the other hand, explicitly account for the geometry and material composition of the propagation environment and therefore require detailed information regarding the environment, relevant antenna patterns, and Tx-Rx geometry. These models have varying levels of sophistication, ranging from two-dimensional (2-D) vertical plane diffraction models (e.g., Longley-Rice, Terrain Integrated Rough Earth Model [TIREM], Variable Terrain Radio Parabolic Equation [VTRPE]) that can be executed on a standard laptop to full three-dimensional (3-D), ray-

traced transmission, reflection, and diffraction models (e.g., X3D and Stingray) (Gribble and Amstutz 2015) requiring a high-performance computer, such as a graphics processing unit (GPU) workstation, to calculate model results.

1.2 Objective

Research led by Dr. Christopher Goodin in the Mobility Systems Branch at the Engineer Research and Development Center's Geotechnical and Structures Laboratory (ERDC-GSL) simulates the physical and onboard sensor performance of remotely controlled robotics in virtual environments, particularly urban scenes. Dr. Goodin's simulation platform is the Virtual Autonomous Navigation Environment (VANE, formerly ANVEL) (Durst, et al. 2012). A limitation of VANE is that is does not predict the fidelity of RF telemetry signals that control the robotics. A need exists for characterizing the RF telemetry channel pertinent to these robotic simulations, particularly in situations when the robot and base station form a NLOS configuration.

The objective of this project was to develop 3-D RF signal propagation modeling capabilities pertinent for robotics remote control and capable of quantifying the channel in urban environments. In future work, modeling capabilities will interface with the VANE platform.

1.3 Approach

This report presents RF telemetry modeling in complex urban environments, including within an actual urban scene utilized by the VANE platform. Empirical solutions for these environments provide fast but potentially inaccurate estimates for propagation losses. Therefore, all models discussed in this report are deterministic in nature. The deterministic models in this report use ray-tracing methods (for 3-D models) and vertical plane diffraction methods (for 2-D models) to predict RF propagation. Ray-tracing methods are considered relatively accurate and computationally fast compared to other propagation methods (Yun and Iskander 2015; Lu, et al. 2010). Specifically, these models use Remcom Wireless InSight (WI) software:

> [WI] is a suite of ray-tracing models and high-fidelity
> EM [electromagnetic] solvers for the analysis of site-

specific radio propagation and wireless communication systems. The software provides efficient and accurate predictions of EM propagation and communication channel characteristics in complex urban, indoor, rural, and mixed path environments (Remcom 2016a).

WI is a type of geographic information systems (GIS) platform, allowing for real-world features (e.g., terrain and buildings) to be tied to actual geospatial coordinates.

This report uses two WI models: X3D and VPUP. Full 3-D models, also called "high-fidelity" models here, are discussed in Sections 2.4, 3.1, and 3.2. These models use X3D code and run on a dedicated GPU workstation in the Signature Physics Branch, ERDC Cold Regions Research and Engineering Laboratory (CRREL). The 2-D models, also called "medium-fidelity" models in this report, are discussed in Section 2.5. These models use 2-D Vertical Plane Urban Propagation (VPUP) model code run via an application programming interface (API) developed for this project. Medium-fidelity models provide propagation predictions with a computational cost more appropriate for real-time simulation or rapid mission-planning applications.

Importantly, the following section presents a workflow for converting VANE scenes into a format that can be imported into the WI platform for RF telemetry signal modeling. In doing so, WI model output can, in principal, be later coupled with VANE robotics simulations.

2 Methods

2.1 Geographic information systems processing

GIS processing plays an important role in this modeling project. Virtually rendered environments—both real and fictional environments—must be carefully reformatted to import into the WI modeling platform. This section documents the process of converting two common types of data into a usable format for WI modeling: first, raw terrain and building Light Detection and Ranging (LiDAR) data, which is generally useful for modeling any real-world environment, and second, mesh object 3-D processing (e.g., Meshlab) data, which is the type of environment data used in VANE simulations.

2.2 GIS Processing: Raw LiDAR to Wireless InSight

The workflow for processing raw LiDAR data uses ENVI LiDAR, ArcGIS, and Quick Terrain Modeler, all of which are interactive geospatial software programs. The workflow is as follows:

1. Collect, georeference, and edit the LiDAR point cloud. Delete large clusters of erroneous points and unwanted items in the dataset, such as cars and trees.
2. Use the ENVI LiDAR module to extract roof footprints. Save roof outlines as 3-D shapefiles (*.shp) that are compatible with Esri and WI products.
3. Edit roof outlines or shapes in ArcGIS.
   a. If LiDAR coverage exists for building rooftops, ENVI LiDAR's Feature Extraction module works although parameters must be tailored to each dataset.
   b. If LiDAR coverage does not exist for building rooftops (as in the case of the CRREL campus LiDAR dataset because it was collected with a terrestrial scanner), manually edit roof outlines by using 3-D GIS software capable of viewing and manipulating 3-D polygon data (e.g., ArcGIS).
4. Create a digital elevation model (DEM) of the ground surface in Quick Terrain Modeler. Use the finalized roof outlines in Quick Terrain Modeler to mask out buildings and other non-ground features. The final ground surface must be in a standard DEM file format (e.g., geoTiff, *.tif) to be imported into WI.

5. Import the ground and building surface into WI. (If the ground is a GeoTiff and the building roofs are shapefiles, then both files are directly importable into WI.)
6. Extrude buildings to the ground by using the built-in WI feature "Extrude to ground" to create walls from the edges of all of the buildings down to the ground.

See Appendix A.1 for information about how to create WI terrain with spatially varying material properties.

2.3 GIS Processing: 3-D VANE models into Wireless InSight

The workflow for processing 3-D VANE models into WI is as follows:

1. A number of 3-D model formats can be directly imported into WI, the recommended format being a COLLADA (i.e., *.dae) file. GSL creates virtual environments from a library of Wavefront (i.e., *.obj) files, which are converted into COLLADA files via Meshlab, a free, open-source 3-D animation software program. This conversion can be done in other computer animation software programs as well, including VUE and Maya.
2. Once the *.obj files are converted to *.dae files, directly import the files into WI as objects (i.e., *.object).
3. To convert the object files to WI proprietary terrain (i.e., *.ter) or city (i.e., *.city) file format, change the tags within the imported ASCII object file. To do this, replace the ASCII tags `begin_<object>` and `end_<object>` with `begin_<terrain>` and `end_<terrain>` or `begin_<city>` and `end_<city>`. Finally, replace the file extension with either *.ter or *.city. This conversion is easily accomplished using a Python script or, for files that are small in size and few in number, by manually changing the tags in a text editor.

Environments generated in 3-D mesh modeling software programs, such as Meshlab, VUE, or Maya, may not have real-world coordinates stored in a way that is compatible with GIS software programs, if such environments have coordinates at all. Because it is a form of GIS platform, WI expects all imported features to have real-world coordinates associated with them. The workaround for this is as follows:

1. First, import coordinate-less features as objects with local Cartesian spatial coordinates (i.e., xyz coordinates; z is the vertical axis) and rotation values about each axis. Upon saving the project, WI automatically creates an object file in the project directory.

2. Convert the object file to a WI terrain or city file by adjusting the associated object text file as explained in the preceding paragraph.
3. Fix buildings to the ground by using the built-in feature "Extrude to ground" via the WI interface. This extends the buildings along the z-axis into contact with the terrain.

2.4 High-fidelity radio-frequency propagation modeling

Full-field solutions, like finite-difference time-domain (FDTD) models, are very accurate but involve intense computational and memory burdens and extended time scales to compute. Modeling RF propagation on tactically relevant time scales in complex terrain necessitates efficient and approximate solutions. The most useful of these approximations includes the important effects of shadowing, reflection, and diffraction within a 3-D environment but necessarily ignores the fine-scale "fast fading" caused by wave interference effects on the quarter wavelength scale (e.g., at 900 MHz, a quarter wavelength is 8.3 cm). Typically, fast fading effects are addressed by including a 10 to 20 dB "fading margin" in the link budget calculation to ensure that the coverage area is modeled conservatively (Sklar 1997).

The high-fidelity modeling in this report uses ray tracing to interrogate the 3-D scene; to determine reflection and diffraction locations within that scene; and to provide outputs of signal power, data throughput rate, and bit error rate (BER) at locations, or along routes, of interest within the scene. The complexity (and accuracy) of the ray-tracing procedure depends on the detail with which buildings, terrain, and their electromagnetic material properties are represented within the scene. Naturally, the desired accuracy of the result must be balanced against the time investment involved in constructing a geometrically and materially accurate representation of the environment, not to mention the time spent on the propagation calculation itself.

The propagation model used for high-fidelity modeling is the X3D model created by Remcom. This model uses GPU processing to speed up the ray-tracing portion of the calculation. Then, from the ray traces and the material properties of the scene, X3D determines the relative power contribution of various reflections and diffractions to a given receiving point within the scene. The software accumulates and sums losses along a given path (due to geometric spreading of the signal, imperfect reflections, absorptions, diffractions, etc.) to determine the RF power delivered by each ray.

The software accounts for the relative time delays and angles of arrival for all signal rays, enabling reasonable estimates on multipath effects at a given Rx location, including delay spread and effect of various antenna patterns on the final received power.

2.5 Real-time radio-frequency propagation modeling

Running a full 3-D ray-traced urban RF propagation model, while much faster than a full FDTD model, still requires significant computational resources and time, depending mainly on scene size and the number of facets used to describe the terrain and building geometries. The significant investment in computational time is justifiable for planning permanent base-station installations within an urban environment but may be impractically slow for real-time robotics mission planning.

In addition to high-fidelity RF propagation modeling, lower-fidelity models, which are less computationally burdensome, are available via a C++ API with the 2-D VPUP model by Remcom. This model type can be executed using an API to produce a simple program callable as one component of a larger, overarching robot mission-planning system. This code ingests urban and terrain geometries along with various Tx and Rx locations and produces an output in terms of received RF power, which is related to the likelihood of successful communication. Because the API is thread safe, multiple Tx-Rx pairs can be processed in parallel using the same urban and terrain geometry, significantly improving speed in cases where many transmit points need evaluation or if multiple frequencies are involved.

Creating an API executable is a straightforward exercise in C++ programming and follows these general steps:

1. Include the necessary libraries to import and process urban and terrain geometries, to expose the necessary RF propagation model functions, and so on.
2. Preprocess the urban and terrain geometries into a format accessible by the API. Preprocessing can take some time depending on the number of facets used but need only be done once for any given scene; subsequent propagation modeling can reuse the processed geometry file. Though the propagation model uses a local Cartesian coordinate system for calculation and output purposes, the origin of the local system can be assigned latitude

      and longitude coordinates to allow easy conversion from local Cartesian (i.e., xyz) coordinates to latitude-longitude-elevation triplets.

3. Assign electromagnetic material properties to the objects within the scene. Material assignment can be specified down to the facet level.

4. Set up Tx and Rx locations within the 3-D scene; specify the center transmit frequency and any relevant antennae patterns.

5. Specify the propagation model to be used, and then calculate the propagation losses between the Tx and all Rx points.

6. Propagation losses can be used raw or incorporated into an RF link budget to determine regions where sufficient RF power is available for the desired communication link. In particular, the propagation loss calculation can be reused in both the base-to-robot link and the reverse robot-to-base link calculation. The base Tx likely is higher powered and uses directional antennae, so the base-to-robot link will generally have a larger useful range than the robot-to-base link.

Once the C++ code is compiled and linked and the model geometry processed and saved, the user can feed in Tx and Rx positions and obtain received power estimates for both base-to-robot and robot-to-base links in real-time. The real-time API code therefore serves as a geospatially aware "RF propagation engine" to support virtual autonomous robotics operations and algorithm testing in any virtual environment.

Appendix A.3 provides API code written to perform real-time modeling of the CRREL campus. Example output from the API code, shown in Figure 1, lists the propagation losses (in dB) existing between the mobile robot platform and the controlling base station and calculates a simple link budget to derive the link margins (i.e., excess RF power above the minimum required for reliable communications) for both the robot-to-base and base-to-robot links. Note that the link margins are not the same because of the (arbitrarily chosen) 9 W transmit power advantage of the base station over the robot. In both cases, the link margins are positive, implying that coverage in this area can be reliably established.

Execution time for this type of API model is 500 ms, which includes considerable overhead in simply loading the scene geometry. Once this initial geometry load is complete, propagation loss calculations can typically be completed in single milliseconds, suitable for real-time simulations or mission planning.

Figure 1. Output of API code for the CRREL campus model, indicating propagation losses and individual link margin values for both the robot-to-base and base-to-robot links.

```
[exx@localhost src]$ ./CRREL-LB
Starting CRREL real time program.
Using Wireless Insite Real Time v2.7.1.2.
Loading preprocessed geometry file.
Loss at point : ( 0.333, -12.19, 125.909 ) --> loss: 98.3152
Robot RX link margin: 31.6848 dBm
Base RX link margin: 21.6848 dBm
Clearing geometry.
```

2.6   General Wireless InSight model workflow

Two test cases, which are presented in Sections 3.1 and 3.2, demonstrate the key capabilities of the WI RF propagation modeling pertinent to RF telemetry simulations in virtual environments. The general WI model workflow is as follows:

1. Assign electromagnetic material properties to the objects within the scene. Material assignment can be specified down to the facet level; but in this work, all buildings were modeled as windowless, homogeneous concrete or metal structures. The ground in both test cases is considered wet earth.
2. Import terrain and object description files commonly used within the robotics performance-simulation community to build simulation scenes (Durst et al. 2012).
3. Assign appropriate electromagnetic material properties to the terrain and objects within the scene.
4. Place antennae and appropriate antenna gain patterns within the scene.
5. Execute the RF propagation model within the scene.
6. Generate the desired model output, such as signal power, data throughput rate, and BER, as a function of position within the scene.

3 Results

3.1 Test Case 1: CRREL campus

The CRREL campus test case leverages an existing extensive LiDAR dataset of the campus, demonstrating that raw terrain and building LiDAR data can be converted into a RF signal model environment. (Recall that Section 2.2 outlines how to reformat this environment type for compatibility with the WI software. Also, the CRREL campus environment makes the comparison of model results to experimental measurements convenient, as discussed in Section 4.2.) The CRREL campus sits on the eastern bank of the Connecticut River in Hanover, New Hampshire, and features a number of connected and stand-alone buildings and a 25 rise in elevation from the riverbank to the eastern side of campus, as shown in Figure 2. Parking lots and brick office buildings dominate the eastern side of the campus while more industrial buildings featuring metal siding and roofs are found near the river on the western side.

Figure 2. CRREL campus modeling environment with Tx 1 through 7, of which Tx 1, 2, and 3 are located on rooftops. An arbitrary driving route through the campus is shown to demonstrate the path and density of measurement Rxs along a typical path. The longest dimension across the entire scene is approximately 450 m.
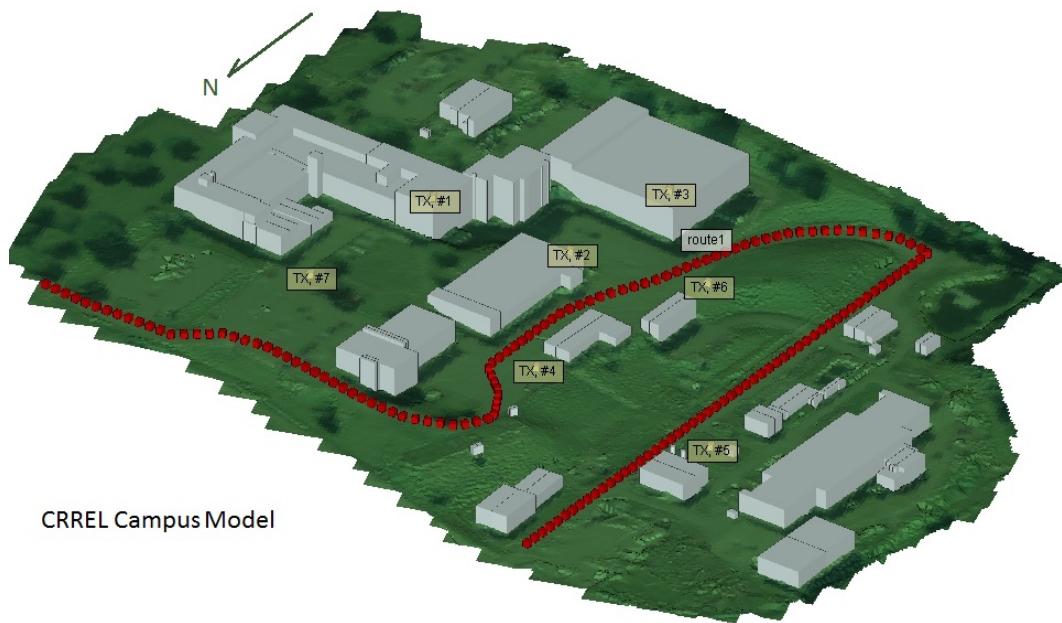
Note that for simplicity, all CRREL campus models ignore the differences in building materials included in the scene, but additional material and geometric detail could be included if desired. All buildings are modeled as

windowless concrete (i.e., ε_r=15.0, σ=15 mS/m, 30 cm thick), while the terrain is modeled as a wet earth (i.e., ε_r= 25.0, σ = 20 mS/m) dielectric half space. The longest dimension across the scene is approximately 450 m.

CRREL campus models include seven different potential Tx sites. Tx 1, 2, and 3 are located in ideal rooftop locations while the remaining Txs are 2 m above the local ground elevation. The Txs use identical half-wavelength vertical dipole antennas and transmit a 4 MHz bandwidth signal at a center frequency of 908 MHz and with a transmit power of 1 W. These signal parameters represent typical robotics telemetry systems capable of real-time video streaming (Sherman 2014), but it is important to note that parameters are easy to adjust for any system configuration. Models account for a maximum of six reflections and one diffraction per propagation path, including interactions with both buildings and the terrain. These settings are defaults in the X3D model.

This section presents several important communications link parameters for urban and terrain geometries:

1. Received signal power, either along a specific path, as shown in Figure 3, or on a grid covering the area of interest, as shown in Figure 4. The received power can be compared with the Rx sensitivity to produce coverage maps.
2. Expected data throughput rates, which vary with modulation type, received power, and background noise levels, and the degree of multipathing present in the environment. See Figure 5 and Figure 6 for data throughput model results.
3. BER, which again varies with modulation, power, background noise, and the degree of multipathing present in the environment. See Figure 7 for BER model results.

Figure 3 shows dominant propagation paths from a ground-based Tx (i.e., Tx 6) to the Rx route introduced in Figure 2. Approximately 105 propagation paths are displayed between the Tx and Rx route. The color indicates the received power via a given path (i.e., the warmer the color, the higher the received power).

Figure 3. The 3-D ray-tracing results for propagation modeling of a ground-based Tx (Tx 6, *circled in red*) to a fixed Rx route through the CRREL campus scene.
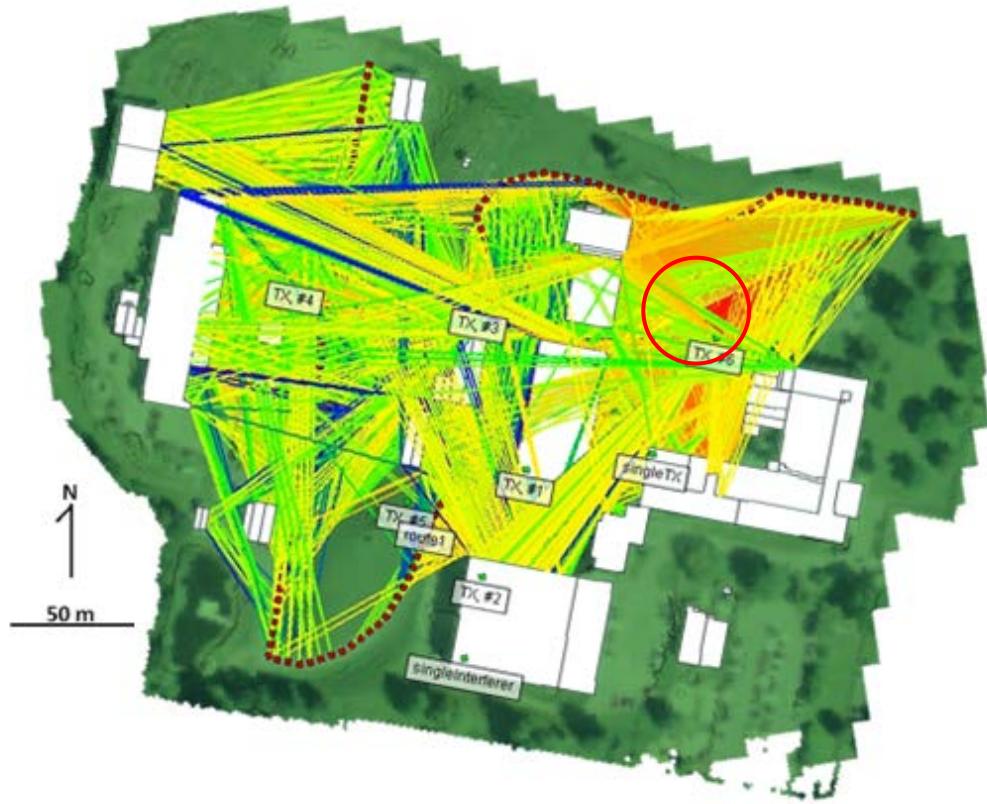
Figure 4 shows the power received from Tx 1 as a function of position throughout the CRREL campus scene. The Tx antenna is omnidirectional and located atop a 20 m tall building, operating at a center frequency of 908 MHz and a transmit power of 30 dBm (1 W). The ray-traced nature of the propagation model is evident from the numerous shadow zones (i.e., purple areas) generated behind buildings and enhanced signal power in front of buildings caused by reflections. Diffraction over rooftops and around building edges contributes power into otherwise deep shadow zones. For high-data-rate robot telemetry applications, the Rx sensitivity is roughly −100 dBm (Anderson et al. 2005), implying that regions highlighted in green and cooler colors would be "dead" zones where the communications link would be lost. The model shows that most of the CRREL campus could be covered by Tx 1, but operation in the very deep shadow zones is not recommended. The model also shows that simply increasing transmission power to 40 dBm (10 W) will not be sufficient to overcome the deep shadowing and establish the link there; at least one additional transmit point would be required for full coverage, probably best located atop a building to the east of Tx 1.

Figure 4. Received power from Tx 1 as a function of position (2 m Rx grid spatial resolution) on the CRREL campus model. The transmit antenna is omnidirectional and located on top of a 20 m tall building, operating at a center frequency of 908 MHz and a transmit power of 30 dBm (1 W).
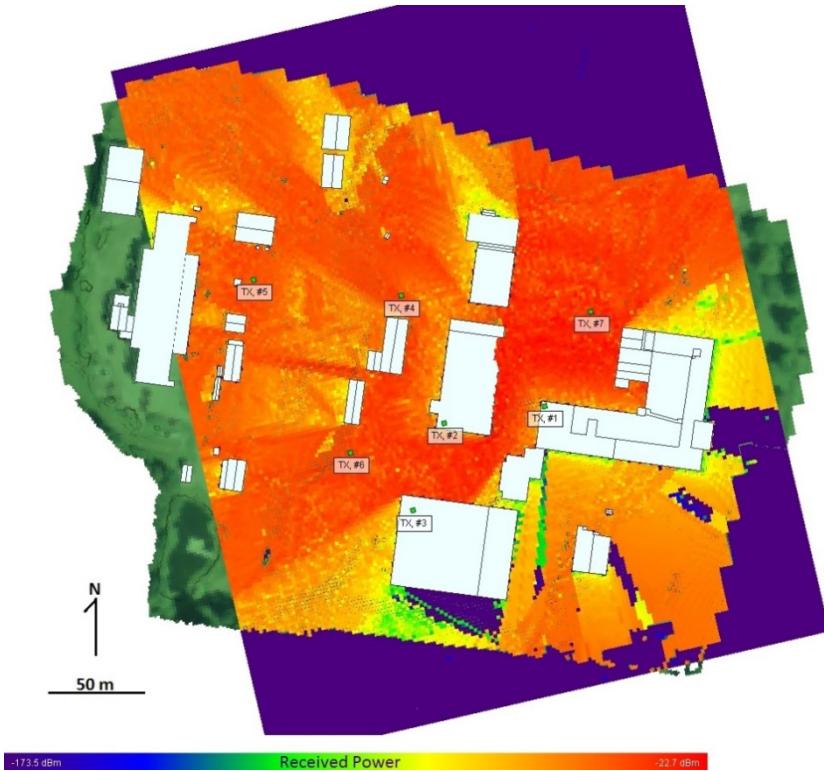
Two data throughput maps, Figure 5 and Figure 6, use all seven Txs to demonstrate the impact of background noise on communication links. RF noise is dependent on both frequency and locale. For instance, if the robotic platforms are to use frequencies in the unlicensed Industrial, Scientific, and Medical (ISM) bands (e.g., 915 MHz, 2.4 GHz, and 5.8 GHz), noise levels may significantly impact robotics operations. Figure 5 shows the results of a low-noise scenario, which involved using only a thermal noise power appropriate for a 4 MHz wide receiving bandwidth, roughly −107 dBm (0.02 pW) (Carr and Brown 1998). No other co-channel interferers are present. This leads to widespread coverage over the CRREL campus at the transceiver's maximum throughput, 2 Mbps, with a few mutually shared shadow zones at the southern and eastern sides of the campus.

Figure 5. Data throughput rate (in units of Mbps) for the CRREL campus model in a low-background-noise scenario. All seven Txs are active in this scenario, giving high-speed coverage over most of the campus. This result is expected as this scenario considers only thermal noise.
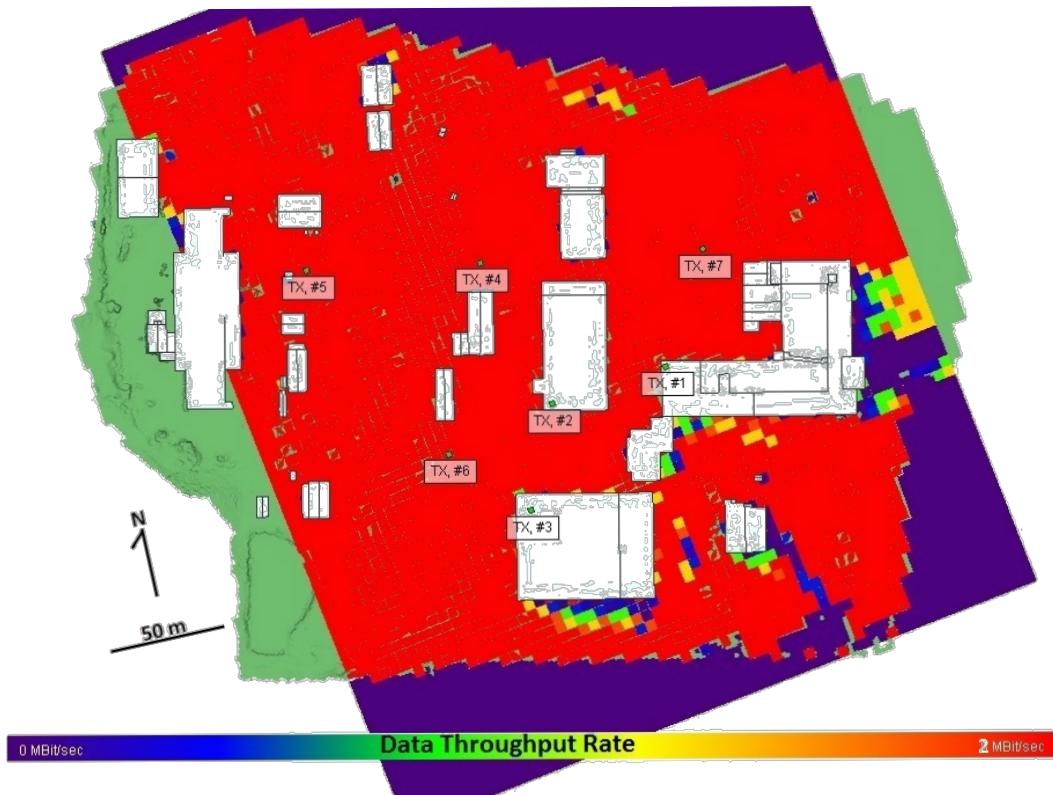
The high background noise (or broadband jamming) scenario considers a 20 dB increase in background noise across the Rx bandwidth, as shown in Figure 6. Even for the relatively small CRREL campus scene, this noise increase causes a noticeable decrease in expected throughput in the scene. In the higher noise scenario, maximum data transfer rates are achieved only within about 50 m of the Txs, and performance drops off rapidly with additional separation and shadowing. This information provides additional insight into mission planning, beyond simply knowing whether the RF link can be established or not. If data-intensive activities are planned in low-data-throughput regions of the scene, these maps allow mission planners to estimate both buffer storage requirements and the transfer time required to move the buffered data back to the base station when the platform moves back into a high-throughput region.

Figure 6. Data throughput rate (in units of Mbps) for the CRREL campus model in a high-background-noise or broadband jamming scenario. All seven Txs are active and yet high-speed coverage exists only within about 50 m of the Txs.
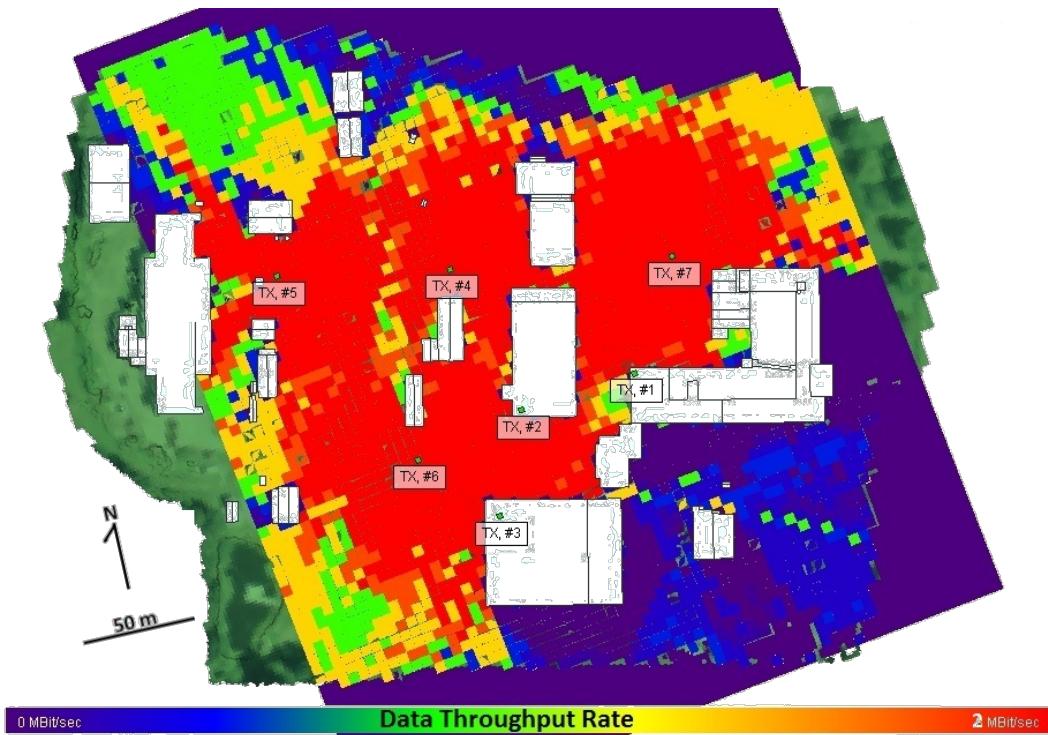
The last important parameter derived from the model is BER, which depends on both the multipathing environment and the presence of co-channel interference. For non-critical data transfers or digital voice communications, BERs of up to 10^{-3} can be tolerated; but for more important data, BERs of even 10^{-8} or lower are necessary to ensure that the rate of errors passed into forward error correction algorithms does not exceed capacity (Freeman 2007).

Bit errors are most often caused by multipath and co-channel interference. Multipath propagation is reasonably well modeled by the ray-tracing technique and is used to estimate the inter-symbol interference component of BER. Co-channel interference can confuse or simply swamp the Rx (depending on the received interference power level) and thus can have a range of effects from an increased BER to complete loss of communications, depending on location within the scene. Figure 7 shows the significant spatial changes in BER for a single Tx (i.e., Tx 1) when a single interferer, equal in power to the desired Tx, is introduced on top of the southernmost building in the scene. In the no-interference scenario, the majority of the campus is covered error-free with limited high BERs located in shadow and diffracted signal zones where signal power is low and/or

longer signal delay spreads associated with multipathing are present. A single co-channel interferer limits the area of viable operation essentially to the intersection of the desired Tx footprint and the shadow zones of the interferer.

Figure 7. Bit error rates (BERs) without (*left*) and with (*right*) co-channel interference for the CRREL campus model. A max BER (i.e., all bits in error) is shown in *red*, while the minimum BER (i.e., all bits correct) is shown in *purple*. The interfering Tx is located on the roof of the southernmost building in the scene. Viable operation (low BER shown in *purple*) in the presence of interference exists at the intersection of the desired Tx footprint and the shadow zones of the interferer.
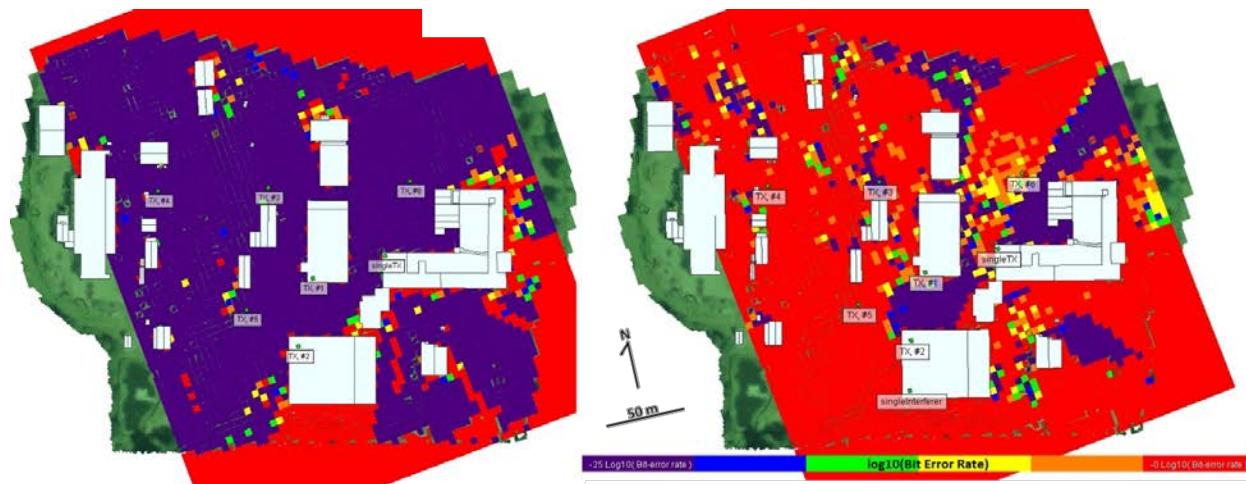
To conclude this section, note that the data throughput and BER calculations both depend on an understanding of the multipath environment, which full 3-D ray-tracing models can achieve with reasonable accuracy. These models are computationally expensive to perform, requiring roughly one hour of computer time with 36 available cores and ray tracing performed by a single high-end GPU. Calculations of received power are less dependent on correctly characterizing the multipath environment and therefore can be done with reasonable accuracy using the real-time models accessible through the API. Thus, received power can be readily calculated in real-time, but throughput and BER calculations require significantly more effort and would also require knowledge of background RF noise levels and the presence, power level, and location of potential interferers.

3.2 Test Case 2: VANE urban scene

The VANE urban scene test case leverages a virtual environment created from 3-D modelling Wavefront files (i.e., *.obj), which are recognized by the VANE modeling platform. (Section 2.3 describes how to reformat this environment type for compatibility with WI.) The scene is fictional, built

on perfectly flat ground, and includes a multitude of high-rise buildings in close proximity, as shown in Figure 8. Spacing between structures creates representative "urban canyon" cells throughout the scene (Kalliola et al. 2003; International Telecommunication Union 2003; Goldsmith 2005). As in Section 3.1, note that this test case ignores differences in building materials for simplicity, but additional material and geometric detail could be included if desired. All buildings are modeled as windowless, thin metal faces (i.e., a perfect electrical conductor), and the terrain is a wet earth (i.e., ε_r= 20.0, σ = 20 mS/m) dielectric half space. (The authors note that wet earth may not be a representative ground material in a built-up area. However, the authors consider the effects of perfectly flat ground to be negligible to the effects of buildings on RF propagation in an urban scene. Therefore, the ground material remained as wet earth for all models.)

The VANE urban model includes three different potential Tx sites. Tx 1, 2, and 3 are all located at rooftop locations. The Txs use identical, omnidirectional, half-wavelength vertical dipole antennae and transmit a 1 MHz bandwidth signal at a center frequency of 2.4 GHz (i.e., within a different ISM band from the previous test case) and with a transmit power of 43 dBm (i.e., 20 W). Again, these parameters can easily be adjusted for any system configuration.

RF propagation analysis performed within this scene uses a full 3-D ray-tracing analysis to establish propagation paths. As in the previous test case, this scenario accounts for a maximum of six reflections and one diffraction per propagation path, including interactions with both buildings and the terrain. Several important communications link parameters can be determined. For brevity, this section considers only two scenarios: Rx signal power (1) along a single ground-level route and a single Tx and (2) using a scene-wide Rx grid and a single Tx.

An arbitrary ground-level Rx route is modeled using a sole Tx atop a 332 m tall high-rise building (i.e., Tx 3), as shown in Figure 9 and Figure 10. The received power via a given path is notated by its color (i.e., the warmer the color, the higher the received power). Note that Rx power is highest when the station is line-of-sight with the Tx along streets and continuous urban canyons, but received power drops off rapidly for NLOS configurations. Figure 10 shows important propagation paths from the Tx to different locations along the Rx route.

A scene-wide, ground-level Rx grid with 5 m spatial resolution and the same elevated Tx (i.e., Tx 3) depicts how a single Tx placed in an ideal location can cast significant signal power into a dense urban scene, as shown in Figure 11. However, it is important to note that deep shadow zones still exist; and as such, multiple Txs are required to provide a reliable link for the entire urban scene, particularly at ground-level locations near or beneath the Tx and at locations that are roughly 500 m or more distant from Tx 3.

A different model, using the same scene-wide Rx grid and a different Tx located atop a 134 m tall structure (i.e., Tx 2), provides a comparison for a lower transmitting height. See these results in Figure 12. (The color scale is identical for both Figure 11 and Figure 12.) The difference is notable, especially the absence of coverage in the "lower-right" corner of the scene. The centrally located, taller buildings in the scene obstruct line-of-sight or primary reflection propagation paths between Tx 2 and the ground in that lower-right area.

Figure 8. VANE urban modeling environment with a multitude of densely arranged high-rise buildings. Clear paths between structures emulate typical urban canyon environments. Buildings have no windows and are modeled as homogenous metal (i.e., perfect electrical conductors). The ground is modeled as a wet-earth dielectric half space. Three Txs (Tx 1–3) are situated atop buildings at varying heights. The longest dimension across the scene is approximately 2.2 km.

Figure 9. VANE urban scene model of a single Tx (Tx 3, *circled in red*) atop a 332 m tall, centrally located high-rise building. Absolute power reaching a ground-level Rx is shown along an arbitrary route through the city.

Figure 10. VANE urban scene model of a single Tx (Tx 3) atop a 332 m tall, centrally located high-rise building. Propagation paths are shown to an arbitrary Rx route through the city.
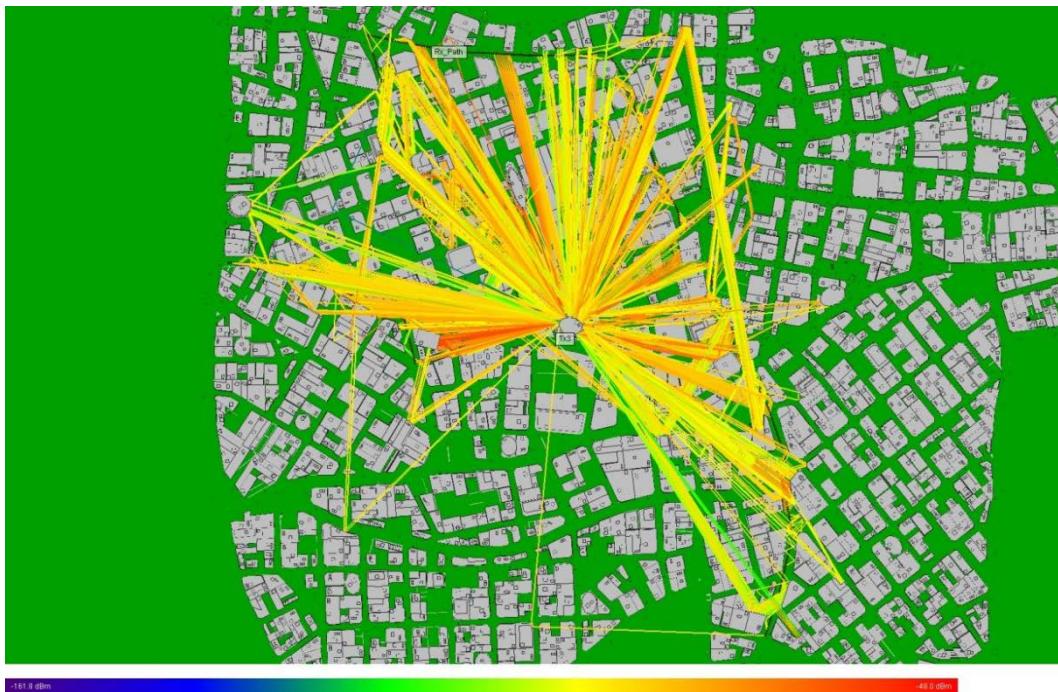
Figure 11. VANE urban scene model of received power for a ground-level Rx grid (with 5 m spacing) and one active Tx atop a 332 m tall building (Tx 3).
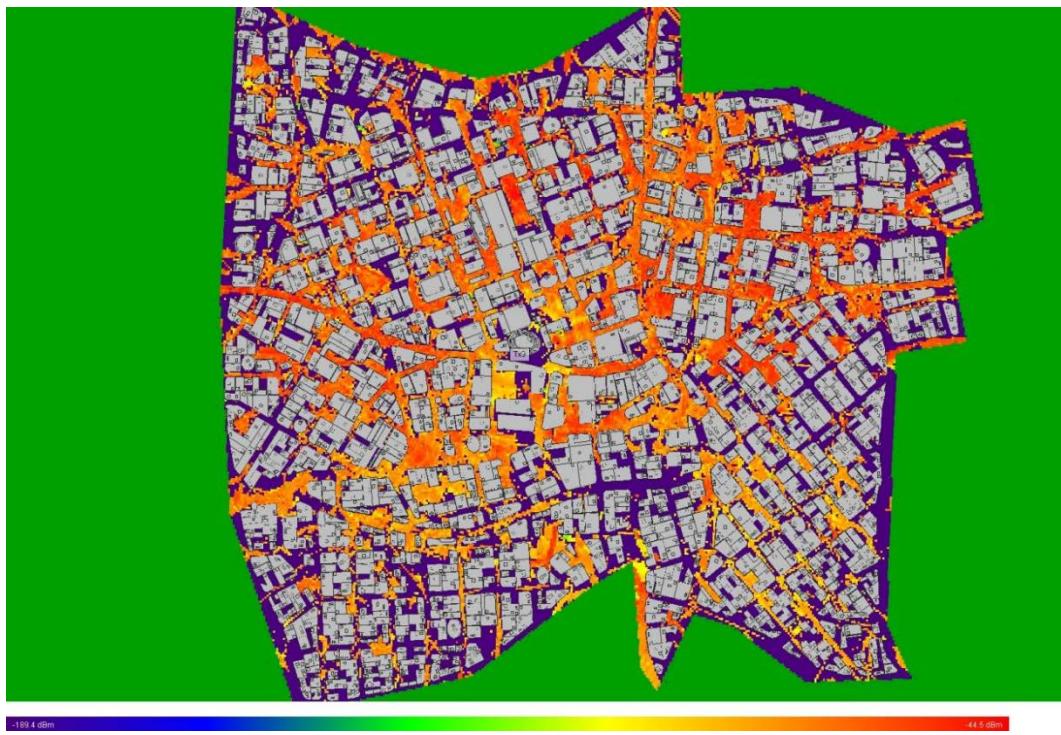
Figure 12. VANE urban scene model of received power for a ground-level Rx grid (with 5 m spacing) and one active Tx atop a 134 m tall building (Tx 2).

To conclude this section, note how signal power is channeled along low-lying, continuous urban canyons, much like a waveguide, in Figure 11 and Figure 12. This phenomenon is unique to dense urban zones and essential to consider in robotics telemetry scenarios when the base station and robot are along a line-of-sight or NLOS urban canyon (Li et al. 2012).

4 Conclusions

4.1 Accomplishments to date

This study implemented high- and medium-fidelity RF propagation ray-tracing models in a virtual rendering of the CRREL campus and in a VANE urban scene. These models demonstrate the ability to transform raw terrain, building LiDAR data, and fictional environments into complete 3-D RF propagation models.

High-fidelity 3-D models, which account for the complex multipathed and delayed behavior of the RF channel, can calculate several important communications parameters, such as received power, BER, and data throughput rate, within any actual or fictional scene. High-fidelity-model results are stored as ASCII files, making them easily imported into external applications, such as the VANE platform. In principle, medium-fidelity 2-D "RF propagation engine" models can supply on-the-fly radio-coverage modeling to external applications, such as VANE simulations. Output data from this model is solely received power, as the 2-D model does not capture the complex multipathed and delayed signal characteristics required for quantifying other communication parameters. Both types of models can simulate an infinite number of Tx-Rx spatial configurations, including bi-static and multiple input, multiple output (MIMO) configurations. The models can also simulate any number of system frequency, waveform, bandwidth, power, and antenna parameters.

4.2 Future work

Through this project, the CRREL research team identified areas of mutual interest between GSL and CRREL, particularly the modeling of RF telemetry signals in complex environments. These areas of mutual interest lay the foundation for future collaboration.

Future work should include comparing model results to experimental measurements of received power and BERs on the CRREL campus. CRREL is well equipped to perform field measurements for comparison to model results. Specifically, an in-house, high-quality, handheld, portable spectrum analyzer (i.e., a Keysight N9344C) could be used to measure absolute power around the campus, in-house MIMO radios could be used for

BER measurements (i.e., Silvus Technologies Streamcaster SC4200 radios), and a mobile channel sounding system recently developed at CRREL (Streeter et al. 2017) could be used to capture power delay profiles, multipathing, delay, and fading data for comparison to high-fidelity model results.

Another component of future work involves providing collaborators access to WI software via the ERDC Research and Development Environment (RDE) network. This would allow others to run WI models conveniently in virtual environments of choice and to use the exact base station and robotic telemetry parameters desired. Telemetry signal coverage information could be smoothly imported for use in VANE by processing the associated ASCII model output files. This solution works for VANE simulations in which base stations and remote robots follow predefined routes or operate in a constant RF environment (e.g., background noise remains constant throughout the simulation). Another avenue to providing collaborators with WI software is via the WI API. By securing access to the WI API for collaborators, medium-fidelity modeling could be incorporated into third-party applications for real-time telemetry signal modeling.

References

Anderson, M. O., S. G. Bauer, and J. R. Hanneman. 2005. *Unmanned Aerial Vehicle (UAV) Dynamic-Tracking Directional Wireless Antennas for Low Powered Applications that Require Reliable Extended Range Operations in Time Critical Scenarios.* Idaho Falls: Idaho National Laboratory, U.S. Department of Energy National Laboratory.

Carr, J. J., and J. M. Brown. 1998. *Introduction to Biomedical Equipment Technology.* 3rd ed. Upper Saddle River, NJ: Prentice Hall, Inc.

Cox, D. C., and R. P. Leck. 1975. Correlation Bandwidth and Delay Spread Multipath Propagation Statistics for 910-MHz Urban Mobile Radio Channels. *IEEE Transactions on Communications* 23 (11): 1271–1280.

Durst, P. J., C. Goodin, C. Cummins, B. Gates, B. McKinley, T. George, M. Rohde, M. Toschlog, and J. Crawford. 2012. A Real-Time, Interactive Simulation Environment for Unmanned Ground Vehicles: The Autonomous Navigation Virtual Environment Laboratory (ANVEL).In *Proceedings of the Fifth International Conference on Information and Computing Science*, 7–10. Liverpool: Conferences Publishing Services.

Freeman, R. L. 2007. Chapter 4: Forward Error Correction and Advanced Digital Waveforms. In *Radio System Design for Telecommunications*, 3rd ed., 175–178. New York: John Wiley & Sons, Inc.

Gans, M. J. 1972. A Power-Spectral Theory of Propagation in the Mobile-Radio Environment. *IEEE Transactions on Vehicular Technology* 21 (1): 27–38.

Goldsmith, A. 2005. *Wireless Communications.* Stanford: Cambridge University Press.

Greenberg, E., and E. Klodzh. 2015. Comparison of Deterministic, Empirical, and Physical Propagation Models in Urban Environments. In *Proceedings of the 2015 IEEE International Comference on Microwaves, Communications, Antennas, and Electronic Systems,* 1–5. Tel Aviv: IEEE.

Gribble, C., and J. Amstutz. 2015. *Effective Parallelization Strategies for Scalable, High Performance Radio Frequency Ray Tracing.* Belcamp, MD: SURVICE Engineering Company.

International Telecommunication Union. 2003. *Propagation Data and Prediction Methods for the Planning of Short-Range Outdoor Radio Communication Systems and Radio Local Area Networks in the Frequency Range 300 MHz to 100 GHz.* Recommendation ITU-R P.1411-2. Geneva: International Telecommunication Union.

Kalliola, K., H. Laitinen, P. Vainikainen, M. Toeltsch, J. Laurila, and E. Bonek. 2003. 3-D Double-Directional Radio Channel Characterization for Urban Macrocellular Applications. *IEEE Transactions on Antennas and propagation* 51 (11): 3122–3133.

Li, Z., R. Wang, and A. F. Molisch. 2012. Shadowing in Urban Environments with Microcellular or Peer-to-Peer Links. In *Proceedings of the IEEE European Conference on Antennas and Propagation*, 26–30 March, Prague, Czech Republic.

Longley, A. G. 1978. Radio Propagation in Urban Areas. In *Proceedings of the 28th IEEE Vehicular Technology Conference*, March, Denver CO, 503–511. Boulder, CO: Institute for Telecommuncation Sciences, Office of Telecommunications.

Lu, J. S., H. L. Bertoni, C. Chrysanthou, and J. Boksiner. 2010. Simplified Path Gain Model for Mobile-to-Mobile Communications in an Urban High-Rise Environment. In *Proceedings of the IEEE Sarnoff Symposium*, 12–14 April, Princeton, NJ.

Parsons, J. D. 2000. Chapter 4: Propagation in Built-up Areas. In *The Mobile Radio Propagation Channel*, 2nd ed., 71–111. New York: John Wiley & Sons Ltd.

Remcom. 2016a. *Wireless InSight User's Guide Version 2.8.1.* State College, PA: Remcom. https://www.remcom.com/wireless-insite-em-propagation-software (accessed 2 May 2017).

Remcom. 2016b. *Wireless InSite Reference Manual Version 2.8.1.* State College, PA: Remcom.

Sherman, P. 2014. FPV for Beginners. *Model Aviation*, April. http://www.modelaviation.com/FPVforBeginners (accessed 2 May 2017).

Sklar, B. 1997. Rayleigh Fading Channels in Mobile Digital Communication Systems Part I: Characterization. *IEEE Communications Magazine*, July: 90–100.

Streeter, S. S., D. J. Breton, and J. M. Corgan. 2017. Spread Spectrum Channel Sounding in a Mountain Shadow Zone. In *Proceedings of the URSI National Radio Science Meeting*, 4–7 January, Boulder, CO.

Suzuki, H.. 1977. A Statistical Model for Urban Radio Propagation. *IEEE Transactions on Communications* 25 (7): 673–680.

Yun, Z., and M. F. Iskander. 2015. Ray Tracing for Radio Propagation Modeling: Principles and Applications. *IEEE Access* 3:1089–1100.

Appendix A: Wireless InSight Supplementary Information

A.1 Modeling heterogeneous ground surfaces

To create different ground-surface types, such as for a ground surface with spatially varying material properties, different DEMs must be created for each surface type. For example, if a DEM contains areas of asphalt and grass, data processing must isolate the soft and hard surfaces in the DEM. Each DEM partition must be imported into WI separately, assigned the appropriate material properties, and then saved as a WI terrain file, a WI proprietary terrain format (i.e., *.ter). Next, the two separate terrain files must be merged into a single terrain file. This is accomplished by concatenating files at the command line by using a simple script (e.g., in Python) or, if the terrain files are relatively small, by copying the terrain information from one terrain file directly into the second terrain file by using a text editor. WI does not perform any surface smoothing or connecting, so any set of DEMs consisting of different materials must be formatted with an identical grid size, spacing, and location to prevent erroneous distortions of the terrain. Splitting a DEM with respect to material properties can be done by importing the DEM into ArcGIS and using the Split Raster feature. The environmental settings in ArcGIS are important to consider when splitting the DEM so that the output maintains the correct coordinates, such as the processing extents, cell size, and snap raster settings.

To create a WI terrain file with two or more material types, parameters in the associated WI terrain file must be adjusted. Specifically, all information between `begin_<Material>` and `end_<Material>` must be moved into the file with different material numbers (e.g., Material 0 = Asphalt, Material 1 = Dry Earth). Then all faces must be moved into the file between `begin_<face>` and `end_<face>` with the correct material number included. These faces must be located within the structure group between `begin_<structure_group>` and `end_<structure_group>`. Again, these steps can be accomplished by a Python script or by directly copying the content in a text editor. See Appendix A.2 for a more in-depth discussion of WI model information data, including terrain, city, object, setup, and diagnostic files.

A.2   Wireless InSight project ASCII files

Every WI model includes ASCII terrain (i.e., *.ter), city (i.e., *.city), object (i.e., *.object), setup (i.e., *.setup), and diagnostic (i.e., *.diag) files that are readily viewed in any text editor. The setup contains detailed information about key model parameters, including terrain, object, and city file references and model type, spatial, material, antenna, Tx, Rx, waveform, and desired output information:

```
begin_<project> CRREL-Propagation
project_id 1432
begin_<globals>
longitude -72.2732290291758
latitude 43.7244005752278
end_<globals>
FirstAvailableUnknownLayoutNumber 0
FirstAvailableStudyAreaNumber 1
begin_<studyarea> studyarea

…
```

After the completion of a model, an ASCII diagnostic file is generated, which documents the progression of the model and any errors encountered. The start of every diagnostic file looks like this:

```
This file contains diagnostic information on the items listed below. Search for
each keyword to go to the relevant sections of the file. Some categories may ap-
pear in more than one place in the file.

1. Version
2. Warning
3. Error
4. Files
5. Receiver Sets
6. Transmitter Sets
7. Antennas
8. Waveforms
9. Materials
…
```

The diagnostic file should be inspected after every model executes to account for any model warnings or errors. Refer to (Remcom 2016b) for additional information on the topic.

A.3   RF propagation engine API code

```
/*******************************************************************
* CRREL real-time propagation -- Daniel J. Breton
* with link margin calculations included
* 23 SEP 2016
*******************************************************************/

// Included for Wireless InSite API functions
#include "../include/uppsInterface.h"

// Included for writing data and error messages to the screen
```

```
#include <iostream>

// Included for string handling
#include <string>

// Define used to switch between creating a .upps file and reading an existing
file
// Switch this to true if you have already created .upps file using this program
#define USE_PREPROCESSED_FILE true

using namespace std;

int main( void )
{
        cout << "Starting CRREL real-time program." << endl;
        upps_Initialize( );
        cout << "Using " << upps_Version( ) << "." << endl;

        unsigned int errorCode;
        string filename;
        string dataPath( "../data/" );
        double longitude;
        double latitude;

        // Sets the longitude and latitude
        longitude = -72.273229029175795;
        latitude = 43.724400575227797;
        errorCode = upps_SetLongitudeLatitude( longitude, latitude );
        if( errorCode )
        {
                cout << "Failed setting longitude and latitude." << endl;
                cout << "Error code: " << errorCode << endl;
                return errorCode;
        }
        if( !USE_PREPROCESSED_FILE )
        {
                cout << "USE_PREPROCESSED_FILE is false. Creating .upps file." <<
endl;

/////////////////////////////////////////////////////////////////////////////////
  // importing urban and terrain geometries
        cout << "Using upps_ImportWICity." << endl;
        filename = dataPath + "CRREL.city";
        errorCode = upps_ImportWICity( filename.c_str( ), MaxZMinusMinZ );
        if( errorCode )
        {
                cout << "Could not import Wireless InSite city file." << endl;
                cout << "Error code: " << errorCode << endl;
                return errorCode;
        }

        cout << "Using upps_ImportWITerrain." << endl;
        filename = dataPath + "CRRELterrain.ter";
        errorCode = upps_ImportWITerrain( filename.c_str( ) );
        if( errorCode )
        {
                cout << "Could not import Wireless InSite terrain file." << endl;
                cout << "Error code: " << errorCode << endl;
                return errorCode;
        }

        // Preprocesses the urban and terrain geometry
        cout << "Completing geometries." << endl;
        errorCode = upps_CompleteGeometries( );
        if( errorCode )
        {
                cout << "Could not complete geometries." << endl;
                cout << "Error code: " << errorCode << endl;
                return errorCode;
        }
```

```
            // Saves the processed geometry file in native UPPS format
            cout << "Saving geometry to CRREL.upps." << endl;
            errorCode = upps_SaveGeometry( dataPath.c_str( ), "CRREL.upps" );
            if( errorCode )
            {
                    cout << "Could not save preprocessed geometry." << endl;
                    cout << "Error code: " << errorCode << endl;
                    return errorCode;
            }
}
else
{
                // Loads the preprocessed geometry file
                cout << "Loading preprocessed geometry file." << endl;
                errorCode = upps_LoadGeometry( dataPath.c_str( ), "CRREL.upps" );
                if( errorCode )
                {
                        cout << "Could not load preprocessed geometry." << endl;
                        cout << "Error code: " << errorCode << endl;
                        return errorCode;
                }
}

// using default materials
 // transmitter #1 location
 upps_Vertex tx;
 double hb = 0;
 double ht = 0;
        tx.x =           62.676;
        tx.y =          -47.641;
        tx.z =           0.0;

 errorCode = upps_GetTerrainHeight(tx,&ht); // ground height in meters above sea
level
        if( errorCode )
        {
                cout << "Could not get terrain height for Tx." << endl;
                cout << "Error code: " << errorCode << endl;
                return errorCode;
        }


 errorCode = upps_GetBuildingHeight(tx,true,&hb); //true ==> meters above ground-
level
        if( errorCode )
        {
                cout << "Could not get building height for Tx." << endl;
                cout << "Error code: " << errorCode << endl;
                return errorCode;
        }

 tx.z = ht+hb+2.0; // sets transmit height 2 m above roof.

 // receiver location
 upps_Vertex rx;
 rx.x =  0.3330;
 rx.y = -12.19;
 rx.z =  0.0;

 errorCode = upps_GetTerrainHeight(rx,&ht);
        if( errorCode )
        {
                cout << "Could not get terrain height for Rx." << endl;
                cout << "Error code: " << errorCode << endl;
                return errorCode;
        }

 rx.z = ht + 2.0; // sets receiver height 2 m above ground-level

        // Calculated loss in dB
        double loss;
```

```
        // frequency (in MHz)
        const double frequency = 900.0;

        // Set the "no data" value
        upps_SetNoDataValue( 250.0 );
        // Calculate loss again at the Rx point
        errorCode = upps_CalculateLossXY( &tx, &rx, frequency, VPUP, &loss );
        if( errorCode == 0 )
        {
                cout << "Loss at point " << ": ( " << rx.x << ", " << rx.y << ", "
<< rx.z << " ) --> loss: " << loss << endl;
        }
        else
        {
                cout << "Could not calculate loss at ( " << rx.x << ", " << rx.y <<
" )" << endl;
                cout << "Error code: " << errorCode << endl;
                return errorCode;
        }

        // Calculate link budgets at this point /////////////////////////
        double Mbr; // base to robot link margin, dB
        double Mrb; // robot to base link margin, dB

        // transmitter powers
        const double Pr_tx=30.0; // dBm. One Watt transmit power for robot
        const double Pb_tx=40.0; // dBm. Ten Watt transmit power for base station

        // gain of both antennas are included in propagation loss calculations

        // assuming BPSK, a desired BER of 1e-6, and a 2.0 Mbps data rate,
        // Eb/N0 is 11 dB ==> SNR = 12.7 * (2Mbps/4MHz) = 6.35 = 8 dB
        const double snr=8.0; // dB, fixed for given modulation type

        // Thermal noise floor at T=290K and B=4MHz bandwidth ==> kTB =-114 dBm
        // Receiver noise figure = 6 dB (internal noise of a good receiver)
        // Overall receiver noise floor = -114 dBm + 6 dB = -108 dBm
        const double rx_noise_flr=-108.0; // dBm, fixed for a given transciever
        // additional noise terms (background noise, broadband jamming, etc) would
        // also be included here.

        // Minimum receive power = receiver noise floor + SNR
        // Minimum receive power = -108 dBm + 8 = -100 dBm
        double rx_min_pwr=rx_noise_flr + snr; // dBm

        // Fade margin of 10 dB, say
        const double fade_margin=10.0; // dB

        // Base to Robot Link margin
        Mbr = (Pb_tx - loss) - fade_margin - rx_min_pwr;

        // Robot to Base Link margin
        Mrb = (Pr_tx - loss) - fade_margin - rx_min_pwr;

        //////////////////////////////////////////////////////////////

        cout << "Robot Rx link margin: " << Mbr << " dBm" << endl;
        cout << "Base Rx link margin: " << Mrb << " dBm" << endl;


        cout << "Clearing geometry." << endl;
        upps_ClearGeometry( );
        cout << endl << endl;

}
```

REPORT DOCUMENTATION PAGE

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
May 2017	Technical Report/Final	

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
High-Fidelity Simulations of Electromagnetic Propagation and RF Communication Systems: T53 Final Report	**5b. GRANT NUMBER**
	5c. PROGRAM ELEMENT NUMBER 602784A

6. AUTHOR(S)	5d. PROJECT NUMBER T53
Samuel S. Streeter, Daniel J. Breton, Michele L. Maxson, and Christopher T. Goodin	**5e. TASK NUMBER**
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
U.S. Army Engineer Research and Development Center (ERDC) Cold Regions Research and Engineering Laboratory (CRREL) 72 Lyme Road Hanover, NH 03755-1290	ERDC TR-17-2

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) USACE
U.S. Army Corps of Engineers Washington, DC 20314-1000	**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

Military Engineering Applied Research

14. ABSTRACT

Newly developed radiofrequency propagation models estimate signal strength, signal coverage, and bit error rates to support mission planning for robotic platforms operating in urban areas. This study involved high-fidelity modeling on a graphics processing unit workstation and included full three-dimensional analysis of reflection, transmission, and diffraction propagation effects within urban landscapes. Real-time propagation modeling is made possible using an application programming interface (API) with simpler, faster models whose output can, in principle, be used for mission planning or platform performance assessment within a virtual scene. This report presents the results of two test cases—within a virtual rendering of the U.S. Army Cold Region Research and Engineering Laboratory campus and within a fabricated dense urban scene—to demonstrate the ability to generate high-fidelity radiofrequency propagation models from building and terrain data derived from (1) LiDAR (Light Detection and Ranging) and digital elevation models and (2) Virtual Autonomous Navigation Environment (VANE) scenes. This report outlines steps necessary to produce lower fidelity, higher speed models using the API and discusses how the API could interface with existing virtual environments and mission-planning tools.

15. SUBJECT TERMS

Channel analysis, Cities and towns, Military robots, Modeling, Non-line-of-sight, Radiofrequency, Radio wave propagation, Ray tracing, Telemetry, Virtual environment

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	SAR	41	**19b. TELEPHONE NUMBER** *(include area code)*
Unclassified	Unclassified	Unclassified			